Контролна работа по ДАА, избираем курс, 27.11.2012г.

Име: ________________________________, ФН:_____, Спец./курс:_______

Задача	1	2	3	4	5	Общо
получени точки						
максимум точки	20	20	15	15	40	110

Задача 1 Подредете по асимптотично нарастване функциите по-долу. Обосновете отговора си и напишете в явен вид подредбата. Предположете, че n е четно.

$$\sum_{i=0}^{n} i(n-i), \qquad n^{\frac{n}{\lg n}}, \qquad \lg((n!)^n), \qquad \sum_{i=0}^{n} 2^i, \qquad n\binom{n}{\frac{n}{2}},$$

$$n^2 \lg n, \qquad \sum_{i=1}^{n} 2^n, \qquad n + \lg(n!), \qquad \sum_{i=1}^{n^2} \frac{n^2}{i}, \qquad \binom{n}{2}$$

Задача 2 Решете следните рекурентни отношения:

а) $T(n) = 3T(\frac{n}{2}) + n$ б) $T(n) = 2T(\frac{n}{3}) + n$

в) $T(n) = 2T(\frac{n}{\sqrt{2}}) + \sum_{i=1}^{n} i$ г) $T(n) = 2T(n-1) + 3T(n-2) + n^2(2^n + 3^n)$

д) $T(n) = 4T(n-2) + n^2 2^n$ е) $T(n) = \sum_{i=0}^{n-1} T(i) + n$

Задача 3 След първото изпълнение на функцията *Partition* от алгоритъма *QuickSort* масивът, подаден за сортиране изглежда така:

$3, 2, 4, 6, 5, 7, 11, 8, 13$

Кой от елементите на масива е бил избран за ключ (*pivot*) на разделянето ? Ако са възможни няколко различни отговора, кои са те ?

Известно е, че ако елемента p е избран за *pivot*, след изпълнението на *Partition* наляво от него ще се разположат числа, по-малки от p, а всички надясно от p ще са по-големи или равни на p.

Задача 4 Известно е, че всеки алгоритъм за разпознаване дали в масив от n числа има поне 2 еднакви извършва $\Omega(n \lg n)$ сравнения в най-лошия случай.

Докажете, че задачата за намиране на двата най-близки елемента (с най-малка разлика по абсолютна стойност) на масив от n числа също изисква поне $\Omega(n \lg n)$ сравнения в най-лошия случай.

Задача 5 Дадени са следните 3 алгоритъма:

```
FibNumA(n:integer)
if n > 1
    return FibNumA(n − 1) + FibNumA(n − 2)
else return n
```

```
FibNumB(n:integer)
create array A[0..n]
if n > 1
    A[0] ← 0
    A[1] ← 1
    for i ← 2 to n
        A[i] ← A[i − 1] + A[i − 2]
    return A[n]
else return n
```

```
FibNumC(n:integer)
create array A[0..2]
if n > 1
    A[0] ← 0
    A[1] ← 1
    for i ← 2 to n
        A[i mod 3] ← A[(i − 1) mod 3] + A[(i − 2) mod 3]
    return A[n mod 3]
else return n
```

5.a (10 точки) Определете сложността по време на трите алгоритъма.

5.b (12 точки) Докажете, че трите алгоритъма изчисляват редицата на Фибоначи.

5.c (12 точки) Ако някой от тези алгоритми бъде стартиран върху 32-битова машина, оценете приблизително за колко голяма стойност на n ще настъпи препълване при сумирането.

5.d (6 точки) С какво третият алгоритъм превъзхожда втория ?